CMPT 280

Tutorial: Heaps

Mark G. Eramian

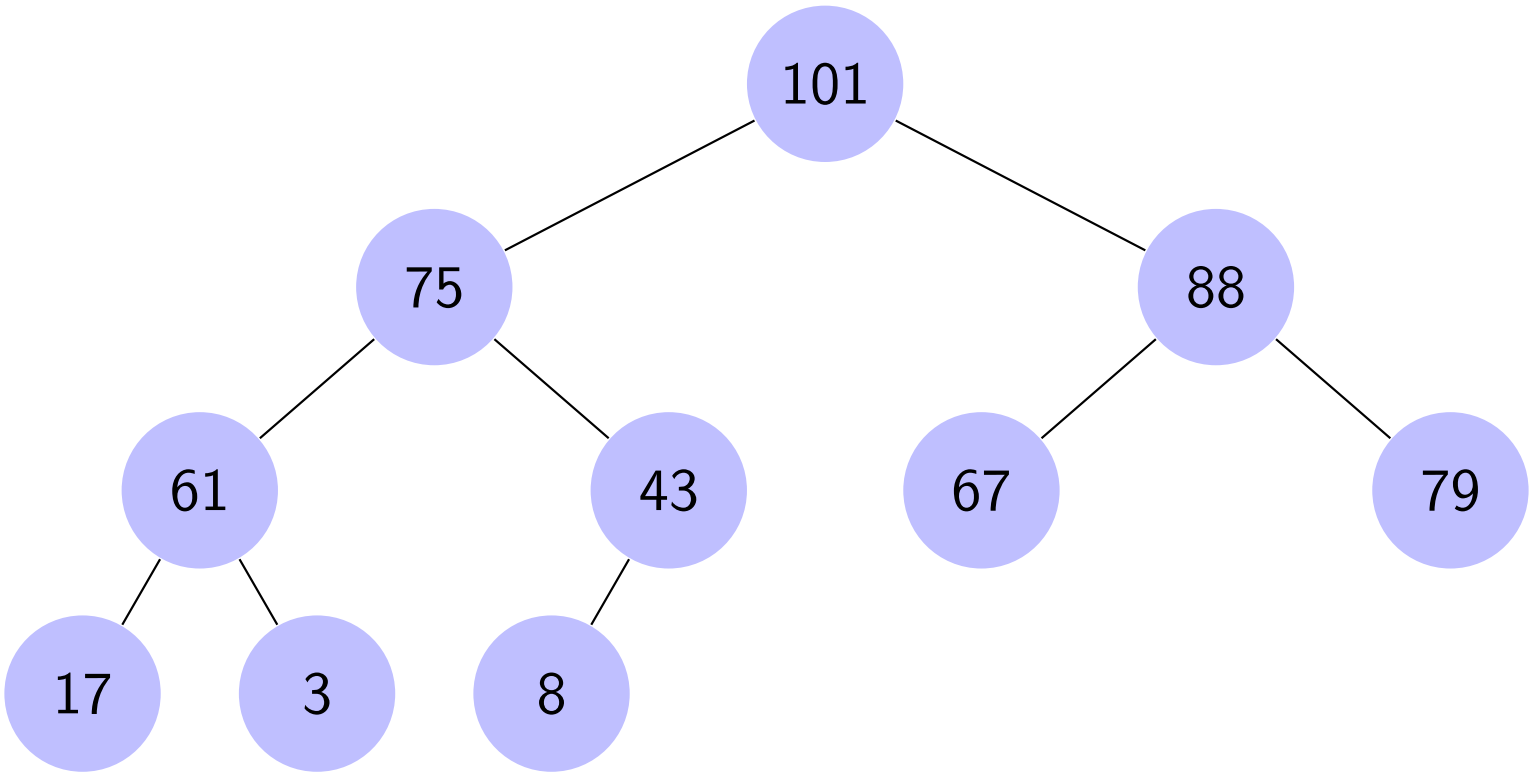
University of Saskatchewan

Heaps

- A *heap* is a special kind of array-based binary tree.
- Recall arrayed binary trees:
  - Every level is full except possibly the last level
  - All elements on the last level are as far to the left as possible.
- Heaps have an additional property, called the *heap property*: **Every node is at least as large as both it's children.**
- Thus, a heap is a "pile" of "stuff" where the largest things are on top.

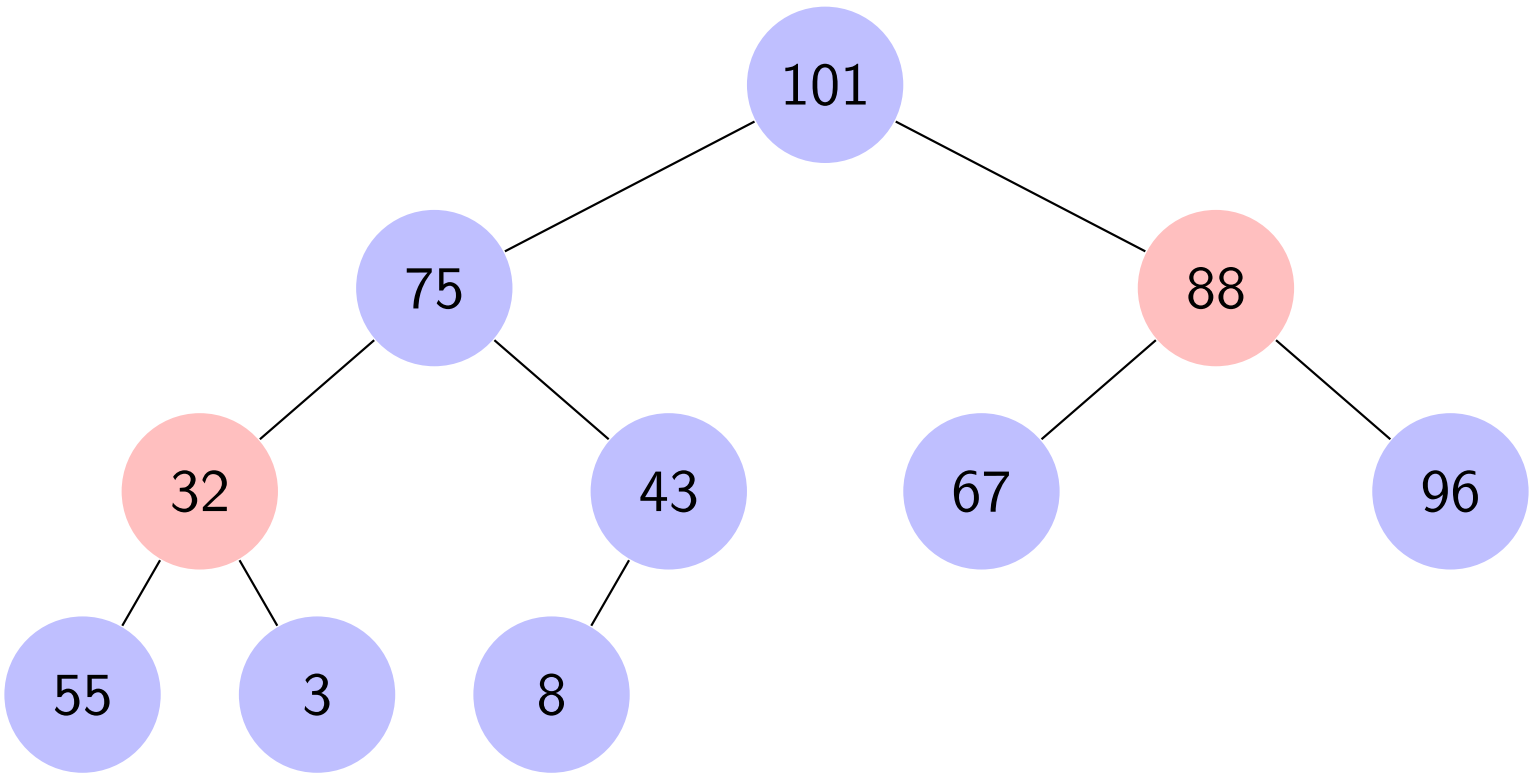
Heap

A heap:



Heap

Not a heap (heap property is violated at red nodes):



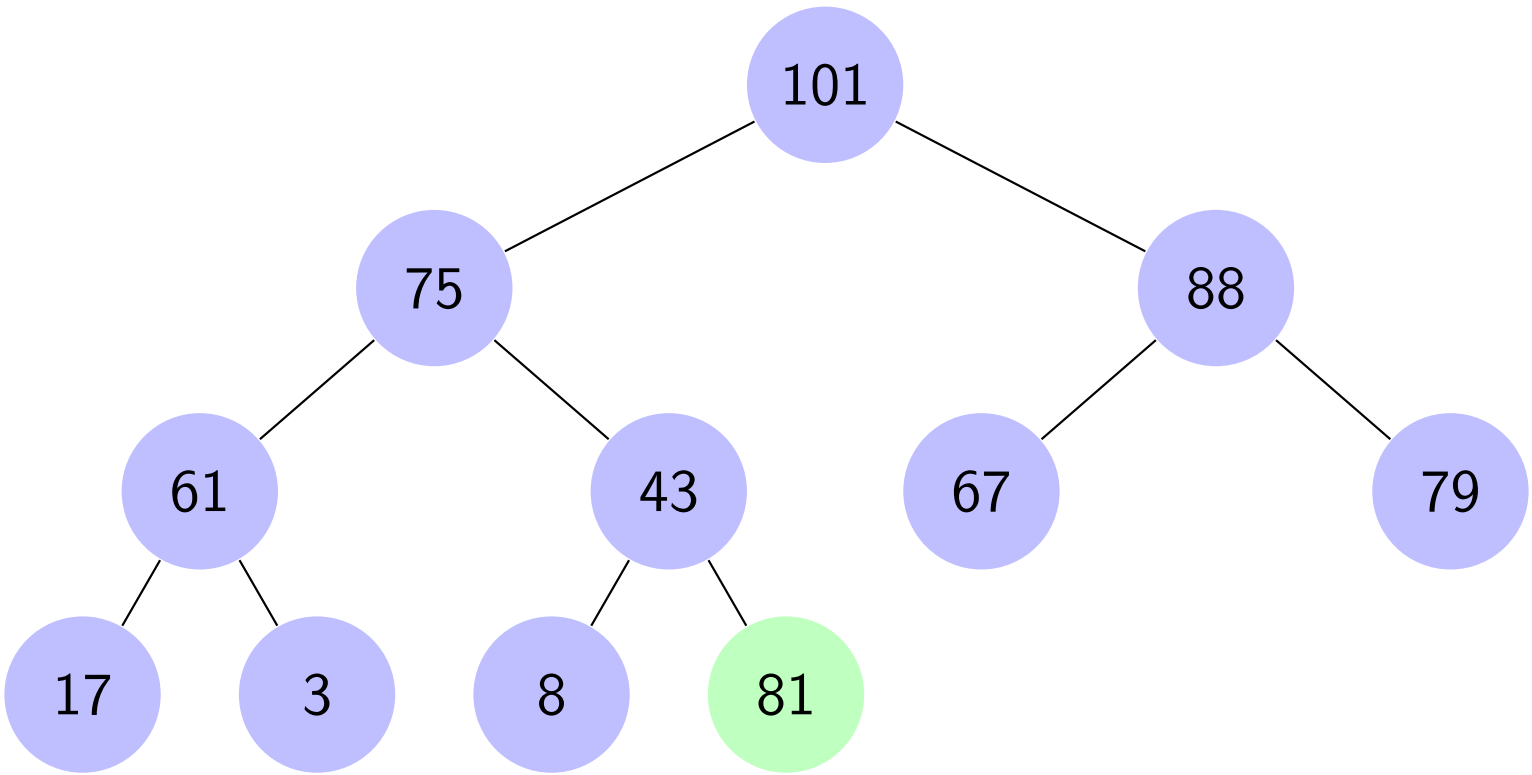
Heap Insertion

Remember: heaps are usually implemented with arrayed trees.

```
1  Algoirthm insert(H, e)
2  Inserts the element e into the heap H.
3
4  Insert e into H normally, as in ArrayedBinaryTreeWithCursors280<I>
5  // (put it in the left-most open position at the bottom level of the tree)
6
7  while e is larger than its parent and is not at the root:
8      swap e with its parent
```

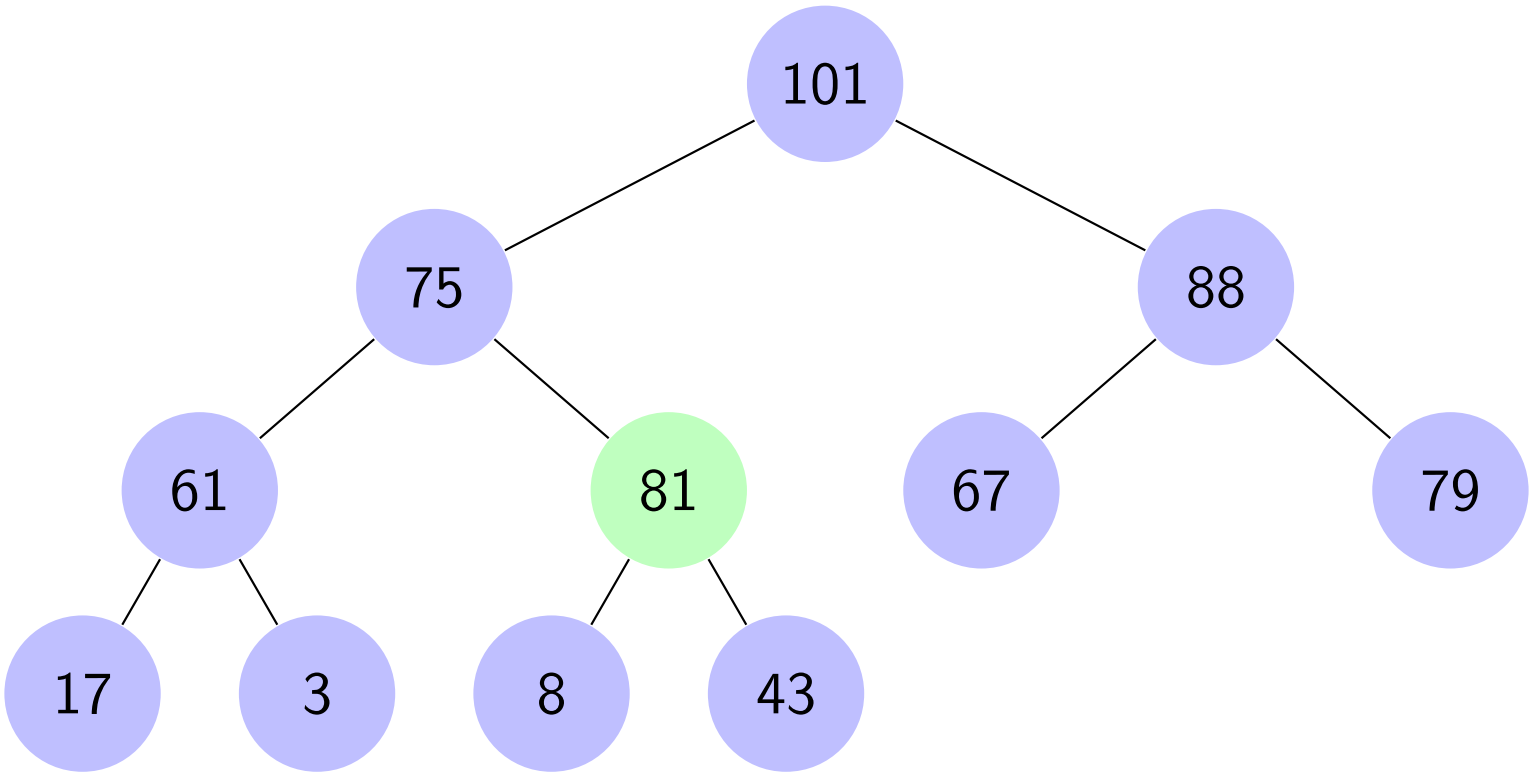
Heap

Insert element 81



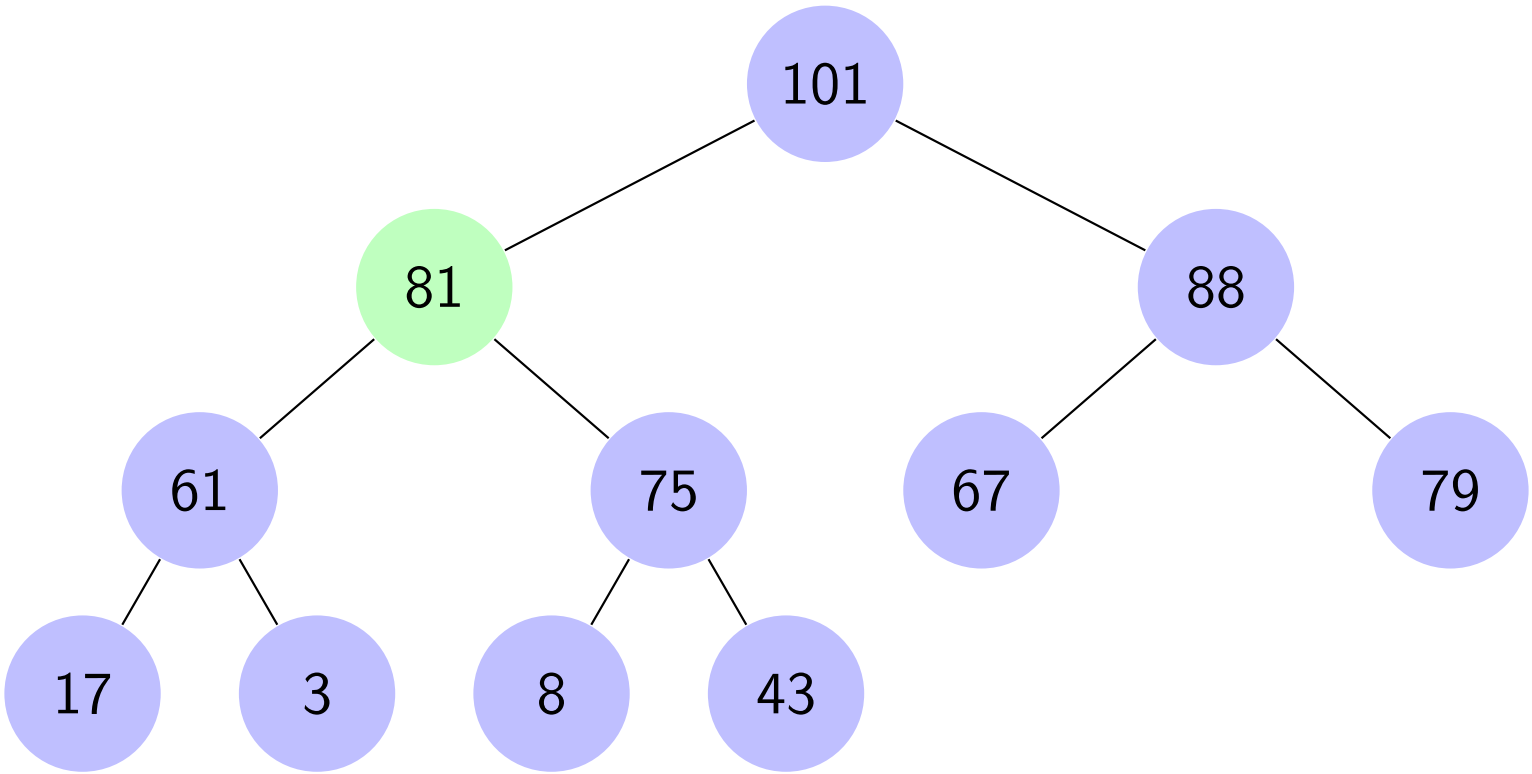
Heap

After first swap.



Heap

After second swap. Heap property now satisfied.



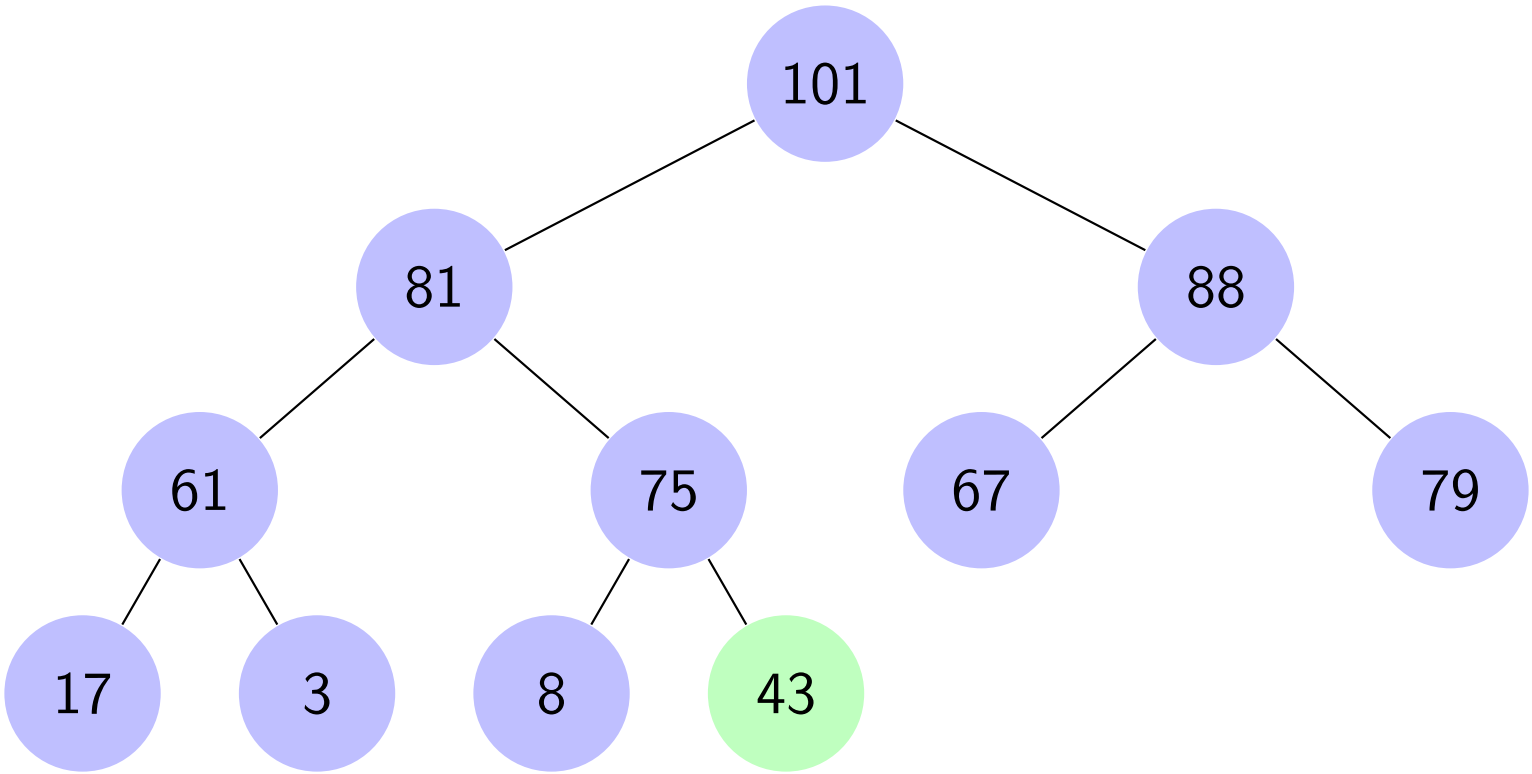
Heap Deletion

Heaps are dispensers. The current item is always the root. Thus when we remove from the heap, we always remove the top item.

```
1  Algorithm deleteItem(H)
2  Removes the largest element from the heap H.
3
4  // Since the largest element in a heap is always at the root...
5  Remove the root from H normally, as in ArrayedBinaryTreeWithCursors280<I>
6  // (copy the right-most element in the bottom level, e, into the root,
7  // remove the original copy of e.)
8
9  while e is smaller than its largest child
10     swap e with its largest child
```

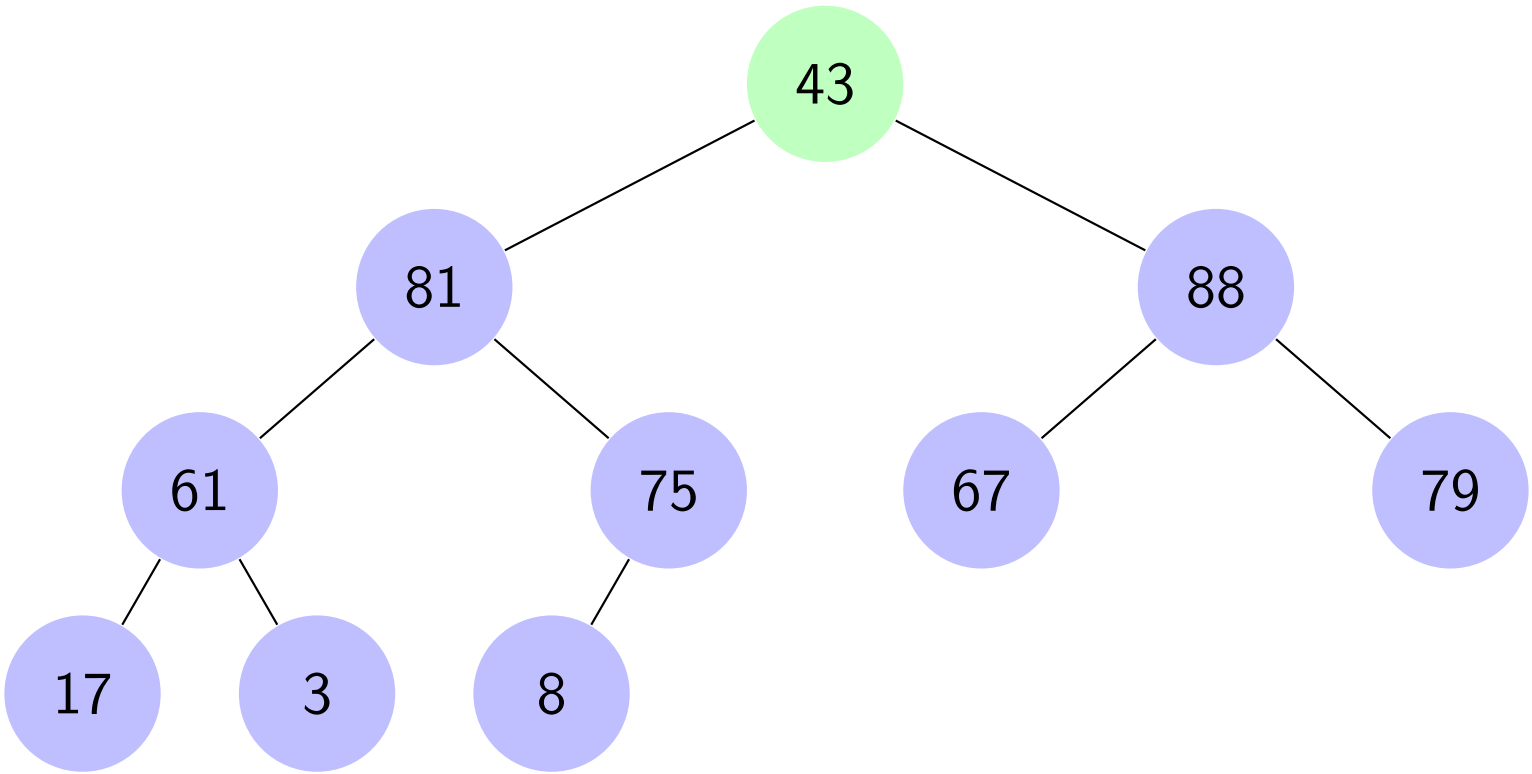
Heap

To remove the current element (always the root), move 43 to the root, overwriting 101.



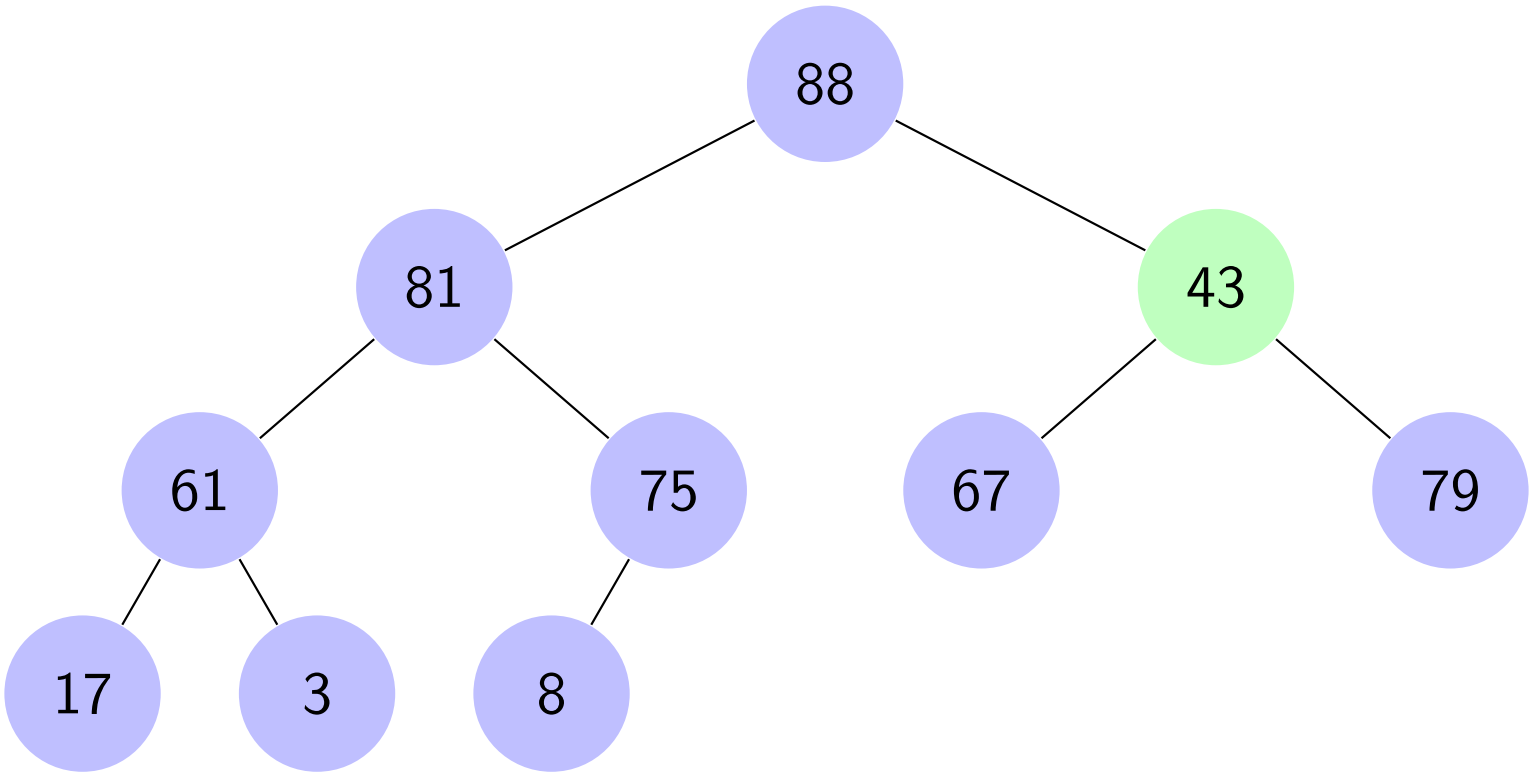
Heap

After move. Now we have to swap...



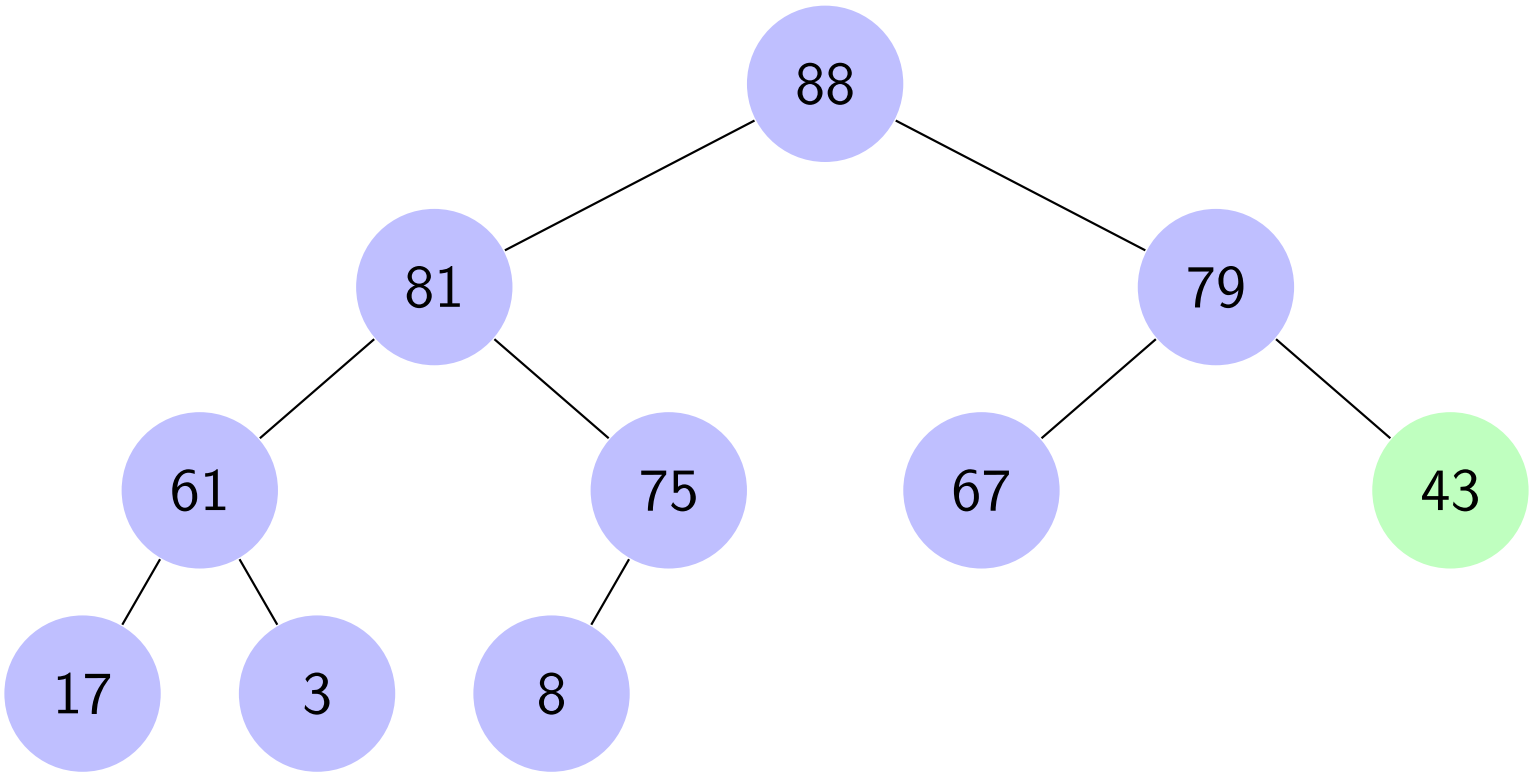
Heap

After first swap...



Heap

After second swap. Heap property restored.



Exercises

- Starting with the heap on the previous slide, insert 3 items of your choice, then remove three items of your choice.
- For at least one of the resulting heaps, draw its array representation.

Data Structures Humour

You are now qualified to read this comic (click here).